AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

<u>LISTING OF CLAIMS:</u>

Claims 1 to 12. (Canceled).

13. (Currently Amended) A method for graphically representing a value of a data type of a formally defined data structure existing as a value tree, comprising:

assigning a window as a graphical user interface to the data structure;

inserting hierarchically at least one generic, scalable, graphical user-interface component in the window, the value tree of the data structure being mapped onto the at least one user-interface component;

providing that the at least one graphical user interface component is in a recognizable relation to at least one node of the value tree;

providing at least one of a graphical representation and a textual representation of the value is selectable for each subtree of the value tree; and

for a processing of the value tree, deriving for each node a value list of all of values compatible with respect to assignment with the data types, and selecting one of the value from the value list for each value assignment; <u>wherein, when compiling the value list, the number of the values to be accepted in the list being restricted in accordance with predefined rules depending on the current context</u>.

14. (Canceled).

15. (Canceled).

16. (Previously Presented) The method of claim 13, wherein a visualization of the window is first undertaken at a time of an initialization of the graphical user-interface and, after that, at least one of data and the value list is initialized, which are derived for a processing.

17. (Previously Presented) The method of claim 13, wherein the value to be represented is transferred in a transfer syntax containing all necessary information for the representation with respect to the data type and the value assignment.

18. (Previously Presented) The method of claim 13, wherein the data type, whose exact type assignment can first be determined at execution time in accordance with a late binding

principle, is inserted as a dynamically changeable subtree in the value tree represented by the graphical user-interface.

19. (Previously Presented) The method of claim 13, wherein for the data type whose exact type assignment is first defined in accordance with a late binding principle at an execution time by a marking of another node, a user is prompted to input information as to whether the exact type assignment should be performed one of automatically and following a manual input.

20. (Previously Presented) The method of claim 13, wherein the value can be transferred from the subtree to another subtree by intermediately storing and clicking on the subtree.

21. (Previously Presented) The method of claim 13, wherein the method is implemented by at least one program module that is integratable in an application program.

22. (Previously Presented) The method of claim 13, wherein additional information to be displayed is storable for each of the at least one node of the value tree which can be uniquely named by a displayed type and a relation to the higher-level type.

23. (Previously Presented) The method of claim 13, further comprising:
continually checking during an inputting of the value of the data type in the value tree to determine whether an input value is permissible for a corresponding data type and to determine whether the input value is identical to a currently active value of the corresponding data type; and
making known to a user a result of the continually checking.

24. (Previously Presented) The method of claim 23, wherein a display format is alterable when the value is inputted before the value is accepted into the value tree.

25. (Previously Presented) The method of claim 19, wherein the marking of the another node includes "ANY DEFINED BY" in a description language ASN.1.

26. (Previously Presented) The method of claim 24, wherein a numerical value is displayed as one of a decimal value and a binary value.

27. (Previously Presented) A method for graphically representing a value of a data type of a formally defined data structure existing as a value tree, comprising:
assigning a window as a graphical user interface to the data structure;

inserting hierarchically at least one generic, scalable, graphical user-interface component in the window, the value tree of the data structure being mapped onto the at least one user-interface component;

providing that the at least one graphical user interface component is in a recognizable relation to at least one node of the value tree;

providing at least one of a graphical representation and a textual representation of the value is selectable for each subtree of the value tree; and

for a processing of the value tree, deriving for each node a value list of all of values compatible with respect to assignment with the data types, and selecting one of the value from the value list for each value assignment,

wherein, when compiling the value list, the number of the values to be accepted in the list is restricted in accordance with predefined rules depending on the current context, and a visualization of the window is first undertaken at a time of an initialization of the graphical user-interface and, after that, at least one of data and the value list is initialized, which are derived for a processing.

28. (Previously Presented) The method of claim 27, further comprising:

continually checking during an inputting of the value of the data type in the value tree to determine whether an input value is permissible for a corresponding data type and to determine whether the input value is identical to a currently active value of the corresponding data type; and

making known to a user a result of the continually checking.

29. (Previously Presented) The method of claim 27, wherein the value to be represented is transferred in a transfer syntax containing all necessary information for the representation with respect to the data type and the value assignment.

30. (Previously Presented) The method of claim 29, wherein the data type, whose exact type assignment can first be determined at execution time in accordance with a late binding principle, is inserted as a dynamically changeable subtree in the value tree represented by the graphical user-interface.

31. (Previously Presented) The method of claim 29, wherein for the data type whose exact type assignment is first defined in accordance with a late binding principle at an execution time by a marking of another node, a user is prompted to input information as to whether the exact type assignment should be performed one of automatically and following a manual input.

32. (Previously Presented) The method of claim 29, wherein additional information to be displayed is storable for each of the at least one node of the value tree which can be uniquely named by a displayed type and a relation to the higher-level type.